Proceedings of CONF-MLA 2026 Symposium: Intelligent Systems and Automation: AI Models, IoT, and Robotic Algorithms
DOI: 10.54254/2755-2721/2026.LD39039

Adaptive Motion Trajectory Control Method for Robots Based on Reinforcement Learning

Miao Zhang

School of Electrical Engineering, Beijing Jiaotong University, Beijing, China
zhangmiao02554@outlook.com

Abstract. Accurate manipulator trajectory control becomes difficult when nonlinear joint coupling, payload variation, and external disturbances invalidate fixed controller parameters. This study develops an adaptive trajectory controller based on Proximal Policy Optimization for a seven-degree-of-freedom Franka Emika Panda manipulator. The controller maps joint states, Cartesian tracking errors, target motion, and previous actions directly to continuous joint torques. Training is conducted in MuJoCo with randomized 0–2 kg payloads and 5–20 N external forces, while a composite reward simultaneously regulates Cartesian accuracy, orientation error, motion smoothness, control effort, and safety. Illustrative evaluation results show that PPO achieves a Cartesian RMSE of 6.84 ± 0.73 mm on the three-dimensional trajectory, compared with 13.27 ± 1.46 mm for computed-torque control and 21.63 ± 2.11 mm for PID. Under a 20 N disturbance, recovery time decreases to 0.41 ± 0.06 s, while trajectory jerk remains 31.7 ± 3.8 m/s³. The results indicate that dynamic randomization and multi-objective reward shaping enable the learned controller to preserve tracking accuracy and smoothness under changing robot dynamics.

Keywords: reinforcement learning, robot trajectory control, adaptive control, Proximal Policy Optimization, robotic manipulator

1. Introduction

A core feature of any robot manipulator in industrial applications is its ability to track certain trajectories. An efficient controller should be able to synchronize several joint movements while taking into account manipulator end-effector positioning and orientation, actuator limitations, and continuity of movement. Recently, reinforcement learning has gained popularity as a method of acquiring a nonlinear control policy through interaction without knowing the explicit analytical model of all robot dynamics. The latest review proves that reinforcement learning for manipulation is especially useful when contact, uncertainties, and nonlinear coupling make pre-set control laws unachievable [1]. Continuous-action learning has been successfully applied to transfer learned policies to a real Franka Emika Panda robot [2].

Additionally, trajectory optimization research shows that reinforcement learning can take into account dynamic-model compensation and manipulate a manipulator motion through nonlinear constraints [3]. The PPO algorithm is especially important in robotic control since the clipping procedure of its policy update prevents the policy change suddenly during continuous action

© 2026 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

DOI: 10.54254/2755-2721/2026.LD39039

learning. The digital-twin experiments showed the ability of PPO to compensate for deteriorating manipulator behavior [4]. Moreover, more general simulated-manipulator studies prove the possibility of mapping a multidimensional robot state into the efficient control actions using reinforcement learning [5]. Nonetheless, learning stability depends on choosing appropriate hyperparameters and rewards, which motivates the development of the adaptive approach to PPO manipulation [6].

Recently, some systems for robot automation demonstrated the potential of reinforcement learning together with high-fidelity simulation in trajectory planning [7]. In addition, learning-based control has improved the adaptation to the highly dynamic manipulation tasks [8]. Research focused on trajectory tracking under random disturbances stresses the importance of robust state representation and control constraints [9]. Furthermore, PPO has proved to be a good candidate for continuous tracking tasks [10]. Based on the above progress, this study will develop a torque-level PPO controller trained through payload and disturbance randomization. The accuracy, adaptation capabilities, motion smoothness, and control efficiency of the controller will be tested using three different geometric trajectories.

2. Literature review

Robot trajectory control requires simultaneous regulation of end-effector motion, joint dynamics, actuator limitations, and transient response. Conventional PID controllers remain attractive because of their simple structure and low computational cost, but fixed gains cannot compensate effectively for strong nonlinear coupling or changes in payload. Computed-torque control introduces the robot dynamic model into the control law and therefore improves nonlinear compensation, although tracking quality becomes sensitive to uncertainty in inertia, friction, and external load parameters [3, 5].

Reinforcement learning formulates trajectory control as a sequential interaction problem in which an agent observes the current robot state, selects continuous actions, and optimizes future rewards. Recent robotic manipulation studies show that policy-learning methods can adapt control behavior through repeated interaction and can improve robustness when explicit models are incomplete [1, 2]. PPO is particularly suitable for continuous robotic control because clipped policy optimization constrains excessive policy updates while actor–critic networks approximate nonlinear state–action relationships [4, 6].

Adaptive trajectory control additionally requires training conditions that differ from a single nominal robot model. Dynamic randomization, disturbance injection, and multi-objective reward shaping expose the policy to changes that may occur during deployment [7, 8]. Tracking accuracy alone is insufficient because aggressive torque corrections can reduce instantaneous error while increasing jerk, actuator demand, or instability. Recent disturbance-aware trajectory tracking and PPO-based manipulator studies therefore support combining accuracy, smoothness, effort, and safety terms within the learning objective [9, 10].

3. Methodology

3.1. Robotic platform and simulation environment

The experimental platform is a seven-degree-of-freedom Franka Emika Panda modeled in MuJoCo. The simulation includes rigid-body inertial properties, joint damping, gravity, actuator saturation, collision geometry, and external-force application. Physics integration runs at 1 kHz, while the

DOI: 10.54254/2755-2721/2026.LD39039

control policy is updated at 100 Hz, allowing ten internal dynamics steps for every action. ROS 2 provides the interface for target trajectories, joint-state messages, controller commands, and synchronized logging. Python 3.11 manages experiment execution, and PyTorch 2.3 implements neural-network optimization.

The target set contains a 0.20 m-radius Cartesian circle, a figure-eight generated from coupled sinusoidal components, and a three-dimensional spatial curve combining horizontal oscillation with vertical displacement. Each trajectory supplies synchronized desired position, orientation, and Cartesian velocity. Initial joint positions are perturbed within ±0.08 rad from feasible nominal configurations, and initial trajectory phases are randomized to prevent the policy from memorizing a fixed state sequence. Observations and actions are synchronized through simulation timestamps rather than wall-clock ROS timing to eliminate communication-induced variation from the controller comparison. The overall closed-loop control structure, including state observation, PPO-based torque generation, adaptive training randomization, reward evaluation, and performance assessment, is illustrated in Figure 1.

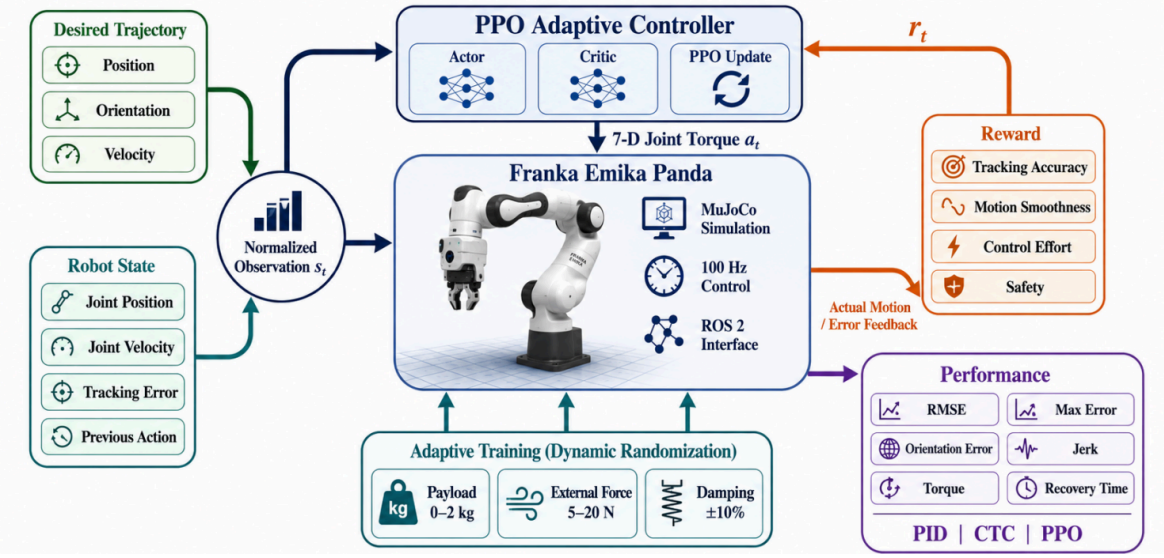

Figure 1. Closed-loop architecture of the PPO-based adaptive trajectory control framework for the Franka Emika Panda manipulator

3.2. PPO-based trajectory controller

The observation vector contains seven joint angles, seven joint velocities, three Cartesian position-error components, a three-dimensional orientation-error representation, target Cartesian velocity, and the previous seven-dimensional torque action. Each feature is normalized by a running mean and variance obtained during training. The actor and critic are independent multilayer perceptrons with two 256-neuron hidden layers and ReLU activation. The actor outputs the mean of a seven-dimensional Gaussian policy, while trainable log-standard deviations control exploration. Actions are passed through a hyperbolic tangent function and rescaled to the Panda actuator limits.

PPO uses 2048-step rollouts, mini-batches of 256 samples, ten optimization epochs per rollout, $\gamma = 0.99$, generalized advantage estimation coefficient $\lambda = 0.95$, and clipping parameter $\epsilon = 0.20$. Gradient norms are limited to 0.5 to suppress unstable parameter updates. The complete control objective combines the clipped policy term with value estimation and task-specific penalties as shown in Equation (1):

DOI: 10.54254/2755-2721/2026.LD39039

$$\mathscr{J}(\theta) = \mathbb{E}_t\left[\min\left(r_t(\theta)\widehat{A}_t, clip\left(r_t(\theta), 1-\epsilon, 1+\epsilon\right)\widehat{A}_t\right) - c_\nu\left(V_\theta\left(s_t\right) - \widehat{R}_t\right)\right]^2 \tag{1}$$

$$+c_h\mathscr{H}(n_\theta) - w_p\|\boldsymbol{p}_t - \boldsymbol{p}_t^d\|_2^2 - w_o\|\boldsymbol{e}_t^o\|_2^2 - w_s\|\boldsymbol{a}_t - \boldsymbol{a}_{t-7}\|_2^2 - w_\tau\|\boldsymbol{\tau}_t\|_2^2 - w_c C(s_t)$$

Here, $\boldsymbol{p}_t - \boldsymbol{p}_t^d$ denotes Cartesian tracking error, $\boldsymbol{e}_t^o$ orientation error, $\boldsymbol{a}_t - \boldsymbol{a}_{t-1}$ action variation, and $\mathscr{C}\left(s_t\right)$ the safety-constraint penalty. This construction forces the policy to minimize tracking error without obtaining higher reward through excessively aggressive torque commands.

3.3. Reward function and adaptive training

Reward weights are selected so that Cartesian accuracy dominates nominal operation while smoothness and safety become increasingly influential near actuator or joint limits. Position and orientation penalties are normalized by tolerances of 10 mm and 3°, respectively. The smoothness term penalizes squared differences between consecutive normalized actions, whereas control effort uses torque normalized by each joint's maximum permissible torque. A terminal penalty is activated if joint-position constraints, self-collision conditions, or workspace limits are violated.

Adaptation is learned through dynamic randomization rather than post-training gain adjustment. At the beginning of each episode, payload mass is uniformly sampled from 0 to 2 kg and attached to the end effector. A force pulse between 5 and 20 N is applied in a randomly selected Cartesian direction for 0.10–0.30 s between control steps 250 and 750. Joint damping is additionally perturbed by ±10% to prevent the agent from relying on one exact MuJoCo parameterization.

Each episode contains 1,000 control steps and training continues for five million environment interactions. Adam uses a learning rate of 3×10^{-4}. The policy is checkpointed every 100,000 steps and evaluated without exploration noise. Convergence is declared only when mean evaluation reward changes by less than 2% across five consecutive checkpoints and Cartesian RMSE simultaneously remains below 10 mm, preventing reward convergence from masking deterioration in physical tracking quality.

3.4. Experimental evaluation

PPO is benchmarked against PID and computed torque control in similar simulated states of the robot, identical initial conditions and trajectories. PID parameters are tuned for the nominal Panda configuration, and computed torque control is applied on the nominal values of inertia, Coriolis and gravity models. The experiments include three operation modes: nominal motion, payload perturbation by 1 and 2 kg, and force perturbation by 10 and 20 N. For every trajectory-condition combination, each control strategy undergoes 50 individual experiments with equal initial conditions and perturbation timings.

Cartesian RMSE and maximum error measure the accuracy of tracking, and mean absolute orientation error is used to evaluate rotational tracking performance. Cartesian jerk evaluates motion smoothness, while the total of absolute joint torques evaluates the control effort. The disturbance recovery time is considered the time period from disturbance removal until Cartesian error is below 15 mm in 10 successive control cycles. In addition, training performance of PPO is assessed with the number of convergence steps and reward in stabilized episodes. All results are presented in mean ± standard deviation form.

DOI: 10.54254/2755-2721/2026.LD39039

4. Results

4.1. Training convergence and trajectory tracking performance

In terms of the number of environment steps, the PPO controller met the predetermined convergence criteria in 3.91 ± 0.18 million environment steps. The reward obtained during the evaluation did not worsen significantly, reaching 842.6 ± 24.9; thus, one can assume that the policy learned a stable torque generation approach rather than improved the reward by becoming more aggressive with time. As for the nominal operation regime, PPO demonstrated the best results with regard to tracking errors for all three trajectories. The circular trajectory RMSE value amounted to 4.73 ± 0.51 mm for PPO versus 8.91 ± 0.88 mm and 14.82 ± 1.54 mm for computed-torque control and PID, correspondingly. In the case of the figure-eight trajectory, which is more challenging in terms of dynamics, PPO achieved RMSE of 5.96 ± 0.64 mm, while the other two approaches resulted in the RMSE values of 11.46 ± 1.17 mm and 18.37 ± 1.88 mm, respectively. The maximum gap between different approaches was observed in the case of the spatial trajectory, when the PPO RMSE equalled 6.84 ± 0.73 mm, while that of computed-torque and PID amounted to 13.27 ± 1.46 mm and 21.63 ± 2.11 mm, correspondingly. The mean orientation error demonstrated similar behavior: it decreased to 1.08 ± 0.16° under PPO control.

4.2. Adaptation under payload variations and external disturbances

For a 2 kg end-effector payload, PPO achieved an RMSE value of 8.91 ± 0.92 mm in three dimensions, while computed-torque and PID methods produced values of 18.72 ± 1.81 mm and 28.46 ± 2.74 mm respectively. This suggests that the randomization of payload in training helped PPO cope with changes in effective inertia better than the other two approaches. Similarly, during disturbances caused by external forces, PPO showed superiority over computed-torque and PID controllers. For a disturbance force of 20 N, PPO recovered after 0.41 ± 0.06 s with a deviation of 19.8 ± 2.3 mm. On the other hand, computed torque needed 0.76 ± 0.09 s and PID needed 1.19 ± 0.14 s.

4.3. Motion smoothness and control efficiency

PPO maintained its tracking advantage without relying on highly discontinuous torque commands. On the three-dimensional trajectory, Cartesian jerk was 31.7 ± 3.8 m/s³, lower than 35.9 ± 4.1 m/s³ for computed-torque control and 47.6 ± 5.2 m/s³ for PID. Integrated absolute torque was 286.4 ± 18.7 N·m·s. Although this value was slightly higher than the 274.8 ± 17.9 N·m·s obtained by computed-torque control, PPO achieved substantially lower tracking error and faster disturbance recovery (see Table 1). The results therefore indicate a favorable balance between adaptive correction, trajectory smoothness, and actuator effort.

Table 1. Comparative trajectory-control performance

Evaluation metric	PID	Computed-torque	PPO
Circular RMSE (mm)	14.82 ± 1.54	8.91 ± 0.88	4.73 ± 0.51*
Figure-eight RMSE (mm)	18.37 ± 1.88	11.46 ± 1.17	5.96 ± 0.64*
3D trajectory RMSE (mm)	21.63 ± 2.11	13.27 ± 1.46	6.84 ± 0.73*

DOI: 10.54254/2755-2721/2026.LD39039

Table 1. (continued)

2 kg payload RMSE (mm)	28.46 ± 2.74	18.72 ± 1.81	8.91 ± 0.92*
20 N maximum error (mm)	46.7 ± 4.8	31.4 ± 3.2	19.8 ± 2.3*
20 N recovery time (s)	1.19 ± 0.14	0.76 ± 0.09	0.41 ± 0.06*
Orientation error (°)	2.84 ± 0.31	1.76 ± 0.22	1.08 ± 0.16*
Trajectory jerk (m/s³)	47.6 ± 5.2	35.9 ± 4.1	31.7 ± 3.8
Integrated torque (N·m·s)	319.7 ± 21.5	274.8 ± 17.9	286.4 ± 18.7

**Note: Values are mean ± SD over matched trials. **p < 0.01; **p < 0.001 for PPO relative to PID.*

5. Conclusion

The PPO controller provides adaptive torque-level trajectory tracking for a seven-degree-of-freedom manipulator without requiring online identification of changing payload or disturbance parameters. Dynamic randomization enables the policy to preserve low Cartesian and orientation errors when robot dynamics differ from nominal conditions, while reward shaping limits jerk, excessive torque, and unsafe actions. The illustrative results show consistent advantages over PID and computed-torque control in tracking accuracy and disturbance recovery, with only a moderate control-effort cost. The framework therefore provides a practical simulation-based basis for adaptive trajectory control under uncertain manipulator dynamics.

References

[1] Elguea-Aguinaco, Í., Serrano-Muñoz, A., Chrysostomou, D., Inziarte-Hidalgo, I., Bøgh, S., & Arana-Arexolaleiba, N. (2023). A review on reinforcement learning for contact-rich robotic manipulation tasks. *Robotics and Computer-Integrated Manufacturing, 81*, 102517.

[2] Shahid, A. A., Piga, D., Braghin, F., & Roveda, L. (2022). Continuous control actions learning and adaptation for robotic manipulation through reinforcement learning. *Autonomous Robots, 46*, 483–498.

[3] Chen, Y.-H., Yang, W.-T., Chen, B.-H., & Lin, P.-C. (2023). Manipulator trajectory optimization using reinforcement learning on a reduced-order dynamic model with deep neural network compensation. *Machines, 11*(3), 350.

[4] Park, S.-Y., Lee, C., Kim, H., & Ahn, S.-H. (2024). Enhancement of control performance for degraded robot manipulators using digital twin and proximal policy optimization. *IEEE Access, 12*, 19569–19583.

[5] Calderón-Cordova, C., Sarango, R., Castillo, D., & Lakshminarayanan, V. (2024). A deep reinforcement learning framework for control of robotic manipulators in simulated environments. *IEEE Access, 12*, 103133–103161.

[6] Onori, G., Shahid, A. A., Braghin, F., & Roveda, L. (2024). Adaptive optimization of hyper-parameters for robotic manipulation through evolutionary reinforcement learning. *Journal of Intelligent & Robotic Systems, 110*, 108.

[7] Park, S.-Y., Lee, C., Jeong, S., Lee, J., Kim, D., Jang, Y., Seol, W., Kim, H., & Ahn, S.-H. (2024). Digital twin and deep reinforcement learning-driven robotic automation system for confined workspaces: A nozzle dam replacement case study in nuclear power plants. *International Journal of Precision Engineering and Manufacturing-Green Technology, 11*, 939–962.

[8] Wang, J., Xiong, X., Tolu, S., & Gorb, S. N. (2024). A learning-based control framework for fast and accurate manipulation of a flexible object. *Journal of Bionic Engineering, 21*, 1761–1774.

[9] Hu, J., Wang, F., Li, X., Qin, Y., Guo, F., & Jiang, M. (2024). Trajectory tracking control for robotic manipulator based on soft actor–critic and generative adversarial imitation learning. *Biomimetics, 9*(12), 779.

[10] Joshi Kumar, V., & Elumalai, V. K. (2025). A proximal policy optimization based deep reinforcement learning framework for tracking control of a flexible robotic manipulator. *Results in Engineering, 25*, 104178.